Survey on Security Issues Techniques Used in Data Warehouses

Mayada J. AlMeghari

Information Systems Department, Faculty of Computers & Information, Cairo University, Cairo, Egypt

Summary
Currently, the exponential growth of the number of Internet users generates huge amounts of databases integrated in Data Warehouses (DWs). The confidential data of business or finance is basic data stored in DWs. Therefore, the problem of keeping data security safe is the biggest risk that faces DWs. This paper shows the common DW security approaches proposed nowadays, covered in two categories. The first category presents some of the approaches ensuring the two major security issues in DWs such as confidentiality and integrity. Because of encryption of a huge amount of query data from DWs making overheads in network, the security of DW plus maintaining high performance has become a necessity for organizations. The second category presents some DW approaches for ensuring data availability by using middleware to obtain high performance. Finally, this paper compares and evaluates the presented DW security approaches of the two categories based on our comparison factors.

Keywords:
Data Warehouse, Confidentiality, Privacy, Integrity, Information Security, Middleware

1. Introduction

Today, in our digital era, there are three main features in business data flow, which are the competitive business environment, organizations need to work in partnership with each other and track their performance for market trend analysis. These features redirect data in order to be collected from more than at least three database sources to generate Data Warehouse (DW). The DW has two specific data properties. First, in DW, there is no necessity for using transaction techniques such as operational application; it has specific database management system (DBMS) technologies. Second, DW operates in read-only mode, so it has specific logical design solutions [1].

DW stores huge amounts of business data, credit card numbers, organization secrets, financial information, personal identity number (PIN) codes, and other personal information. So, these data are vulnerable to attackers' desires [2]. The data stored in DW needs to be transformed to be unreadable to attackers, such as the encrypted data that is generated from data encryption algorithms. This is the main reason for security problems in DW systems. For example, Advanced Encryption Standard (AES) technique in Oracle 10g edition provides strong encryption. However, the encryption process needs an additional storage space for encrypted data and extends query response time. Data security in DWs is very challenging [3], [4].

This survey paper presents many DW security approaches that have implemented and designed security issues in DWs. As mentioned clearly in information security, the security issues contain three attributes known as the triad CIA of Confidentiality, Integrity, and Availability [5]. Confidentiality is used to protect data from unauthorized disclosure. Integrity is used to prevent unauthorized users from changing data. Availability guarantees that data is accessible to authorized users all the time [6]. To cover the three security issues, this paper collects the DW security approaches in two categories. In the first category, some approaches have achieved two security issues in DWs through ensuring confidentiality and integrity of data. In the second category, other approaches have achieved high performance through ensuring data availability by using middleware.

The remainder of this paper is organized as follows: Section 2 presents many approaches that apply two security issues in DWs, which are confidentiality and integrity in the first category. Section 3 provides some of DW high-performance middleware approaches ensured data availability in the second category. Section 4 presents a comparative analysis of DW security approaches for the two categories. Finally, Section 5 offers our conclusion and future work.

2. DW Security Approaches as Confidentiality and Integrity

A DW stores a huge amount of sensitive data, which means large distributions over the internet. Therefore, it is important to search for approaches that apply security issues in DW environments. Some of them focused on ensuring data confidentiality by applying encryption process. Other approaches ensured data integrity by applying hash functions and digital certificates.

A. Deshmuk and R. Qureshi [7] presented the Transparent Data Encryption (TDE) technology for encrypting databases on different places. This technology provided transparent, standards-based security to protect data on network, disk or backup media. It can be used to grant high levels of security to database files storing other

Manuscript received March 5, 2017
Manuscript revised March 20, 2017

information that requires protection. Microsoft SQL Server 2008 used TDE to encrypt database content by creating a Master Key. Thus, TDE involves creating master key, and ensuring protection by the certificate and methods to set the database to use in Microsoft SQL Server 2008 encryption. The Service Master Key is created in accordance with SQL Server 2008 setup. Then Data Protection Application Programming Interface (DPAPI) encrypts the Service Master Key. This Key encrypts Database Master Key for master database. Database Master Key is used to create the Certificate. This certificate encrypts the database encryption key in the user database as shown in Fig. 1. The whole database is safe by the Database Master Key of the user through using TDE technology.

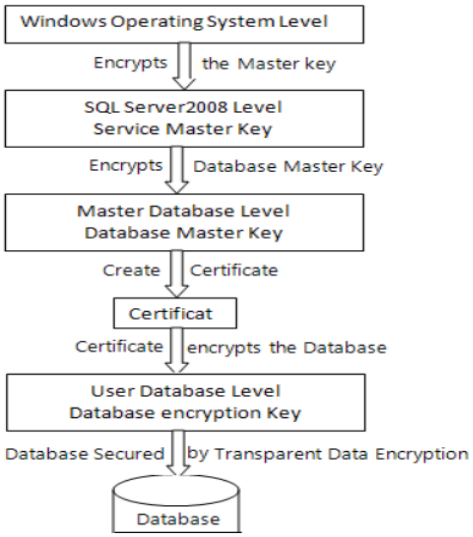

Fig. 1 Microsoft SQL Server 2008 TDE

In 2015, V. Attasena et al. [8] proposed a new multi-secret sharing approach for DWs in the cloud and allowing on-line analysis processing. They designed and implemented a family of encryption methods that enforce data privacy, integrity and availability. This approach minimized shared data size and used the sharing for a DW as OLAP queries run on shares. They developed two schemes: Scheme-I transforms data into blocks to efficient computing and storage costs. This scheme provides two types of hash signatures. Inner signature is used to verify data correctness in all data pieces of block. Outer signature is used to verify the correctness of shares. Scheme-II permits sharing and querying DW in the cloud. Each attribute value in each row is shared independently in relational database. Each primary key is replaced by an unencrypted sequential integer key. All types of attribute are transformed into integers, but Booleans are not encrypted in order to save computation and data storage costs.

A Mixed Cryptography Database (MCDB) framework is proposed to encrypt databases over un-trusted networks [9]. They presented the design of semi-trusted databases to protect sensitive data. In the client-server based encryption, the data is encrypted using either a server key in a trusted database, or a client key in a non-trusted database. In the MCDB framework, the client sends a query request (Q) to many databases servers as shown in Fig. 2. The trusted third party matches the query requests and distributes them (Q1, ..., Qn) to the dependable servers. Then, the query results are merged before submitting the final result to the client. This approach addressed database security issue and contributed significantly in strengthening the protection of sensitive data even if the database server is attacked at multiple points internally or externally.

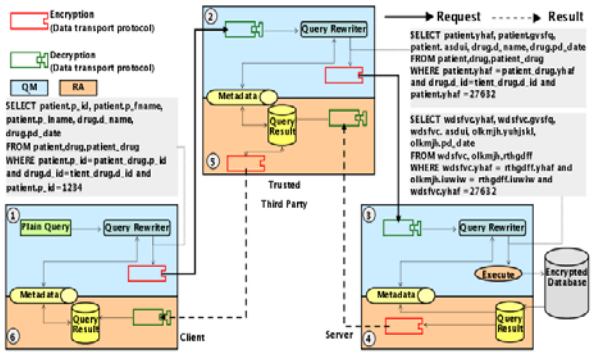

Fig. 2 MCDB architecture

T. Ge and S. Zdonik [10] proposed a new light-weight encryption method called Fast Comparison Encryption (FCE) for the column stored in DWs with trusted servers. The low overhead of FCE makes the comparison fast and efficient. They provided powerful proof that FCE is as secure as any original block cipher through using Information Chosen Plaintext Attack Database (INFO-CPA-DB "a relaxed measure of security "). To prevent the attacker from accessing the values in the memory of the database software, they have separated the security issue of the communication channel between the client and the DBMS server, and the issue of the security of the on-disk data as shown in Fig. 3. Thus, data security in the storage system of the server represents a divided and conquered approach for DBMS security. This system is aimed at the encryption to ensure the security of the on-disk data.

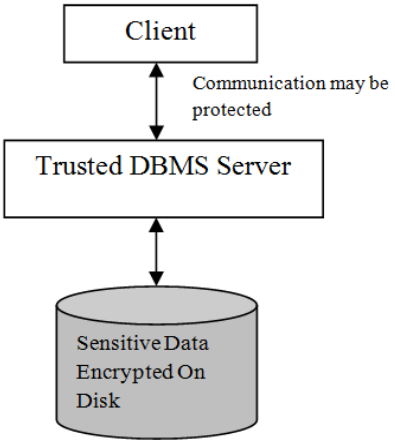

Fig. 3 System model

This approach is firstly considered a version of FCE based on random permutations called r-FCE, which is a symmetric key encryption scheme for a DBMS. The authors presented the FCE algorithm for C-Store by replacing the random permutations of r-FCE with a secret k-wise independent function. FCE acquires low overhead in encryption process because it uses any block cipher to encrypt only a few bytes of random seeds in each page and uses lighter-weight computation to encrypt data in a page. Also, FCE has lower decryption overhead because it uses early stopping mechanism, such as compression method.

In [11], a novel database encryption scheme is proposed for enhancing data sharing inside a database. This scheme provided high-security performance and easy sharing of the encrypted data through combining the conventional encryption and the public key encryption. It also provided secure storage for data security and effective key management. This approach aimed to solve the problems appeared in traditional database encryption models. In the server side, database administrator may misuse his authority, or an attacker can get the privileges of the administrator and cause harm. Also, in the client side, keys protection is difficult to be kept in a secure place. Thus, no database security can be ensured. The environment of this approach is found in an Oscar commercial DBMS for protecting sensitive data and enhancing security requirements. The scheme architecture provided a secure environment to prevent unauthorized access to important sensitive data. This approach made a combination of the conventional encryption and the public key encryption in the database server for the private keys to be presented in the client side. Each user has a key pair (a public key and a private key). In addition, a Certification Authority (CA) is provided to guarantee the real identity related to a public key. The public keys are stored in a table in the security catalog. Then, they are encrypted with the database master key by using a fast conventional encryption algorithm. In this scheme, the private keys are isolated with the public keys and the encrypted data.

In [12], the authors wrote the guidelines for data encryption and data integrity, which is called Hash Security Module Encryption Strategy. This approach heaved a state of the art algorithm and a mode of operation (repetitive patterns, updates, huge volume of encrypted data) to be used. This approach developed a database encryption strategy, which can perform the encryption in three levels as shown in Fig.4: a) Storage-level encryption that is used to encrypt data in the storage system to protect the data at rest. b) Database-level encryption, which is part of the database design, and can be done in a row, a column, or a table. c) Application-level encryption, which performs data encryption within the application that provides the data into the system. The authors presented two solutions for key management to prevent the disclosure of encryption keys. The first solution is a hardware Security Module (HSM) used to provide a secure storage for encryption keys. The other solution is a Security Server approach that manages users, roles, privileges, encryption policies and encryption keys.

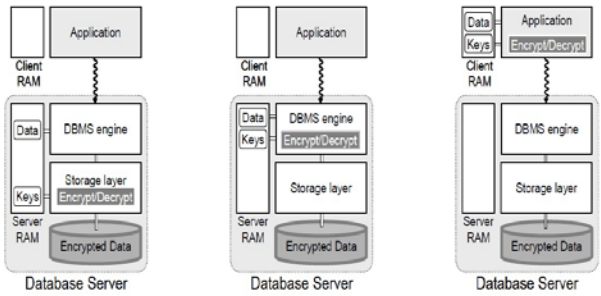

Fig. 4 Database encryption levels

Reddy et al. [13] proposed an encryption scheme to preserve the data type of the plaintext resource. This scheme aims to minimize the need for changes to database structures through preserving the data type of the encrypted field. It involves that each encryptedtext field is as legal as the plaintext field it replaces. This scheme defines the suitable alphabet of correct characters and performs all operations within the constraints of the defined alphabet. Each different data type requires a sensible choice of alphabet. Each character in the plaintext string is replaced by its corresponding position or index as integer number. This integer is between zero and one, and it is less than the total number of characters in the alphabet. If a plaintext character is an incorrect alphabet, it is copied to the output and removed from the string that is to be

encrypted. The authors used modular addition to confirm that generating correct characters only.

3. DW Security Approaches as Confidentilaity and Availability for High-Performance Using Middleware

The large scale of users in the internet networks that have access to the DW is the first challenge for the available system in client- server based. The query result in DW systems shows a huge amount of data compared to the query result in database systems. To obtain a stabled system in the web environment, the network must be available for authorized users without the obstacle of request timed-out. Many DW security approaches can ensure the availability of data to reach high performance as presented below:

R. J. Santos et al. [14] proposed an encryption approach in DWs called a Specific Encryption Solution tailored for Data Warehouses (SES-DW). This approach is applicable in any DBMS with any CPU features, and it requires small storage space and computational efforts. Thus, their approach showed better database performance than other encryption solutions and provided substantial security strength. There are basically two forms to encrypt data in DBMS: column and tablespace encryption. The SES-DW uses the column form for encrypting data to avoid storage space and computational overhead. The authors used a middleware to apply their encryption/decryption methods in order to ensure that the data is securely and fast processed. As shown in Fig. 5, the data flow between user/client and SES-DW middleware can be presented in three steps. First, user's actions must go to the security SES-DW middleware application to get results. Second, the middleware application receives and parses the instructions, fetches the encryption keys, rewrites the query, sends it to the DBMS for processing and retrieves the results. Finally, those results return to the user's application.

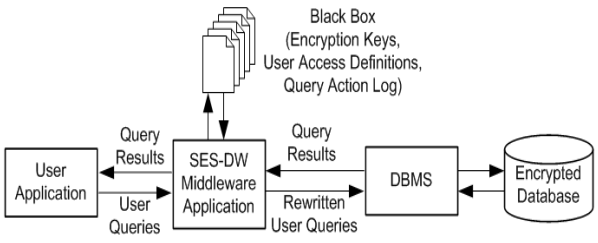

Fig. 5 Architecture of SES-DW

In [15], the authors proposed the distributed mining algorithm, which is the distributed version of Apriori algorithm. Their system is integrated with a set of algorithms, such as the distributed mining algorithm, K&C algorithm and AES algorithm. The main goal of this is to reduce the amount of information that is revealed about the private database through the sites using encryption mechanism. This approach can handle huge data sets with speed through utilizing available resources in the distributed system. This approach protects the individual transactions information in different databases at each site as shown in Fig. 6. The AES encryption algorithm is used to increase the confidentiality. While the K & C algorithm is used for integrating lists of locally frequent itemsets. This approach is more efficiently of distributed association rule mining by security assumptions and strong rules.

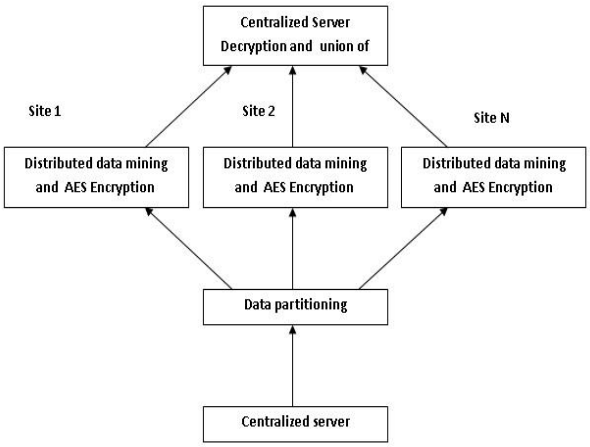

Fig. 6 Secure distributed database

A MOdulus-BAsed Technique for Data Masking in DW (MOBAT) is proposed in [16]. This technique aims to present a specific practical column masking solution for guaranteeing secrecy of privacy in DWs. It is based on a masking formula with two modulus (division remainder) and two simple arithmetic operations. The masking formula ensures that sensitive data is replaced by realistic but not real data. This formula prevents access to real original data in case the attacker can access and retrieve data from database. The authors used MOBAT Security Application (MOBAT-SA) which acts as a middleware between the masked database and the user application. The MOBAT depends on three masking keys: two of them are private and the other one is public. These keys are encrypted and stored in the black box. This approach provides strong data security in DWs with low overheads in storage space and query response time.

In 2015, Preeti and K. Khatka [17] presented a Web-based operating system (WebOS) which is a new form of Operating Systems. This system used the desktop as a virtual desktop on the web, accessible by a browser, with multiple integrated built-in applications that allow the user

to manage and organize his/her data from any location without difficulty. They proposed the design of Transposition-Substitution-Folding-Shifting encryption (TSFS) algorithm to enhance security. TSFS algorithm provides a strong security to the databases at less time cost for only encrypting and decrypting sensitive data fields, such as passwords and contact numbers. This algorithm supports the encryption of special characters. Moreover, it improves performance without compromising either query processing time or database size.

In [18], the security of the DW has been increased through design and implementation of security mechanism. It enhances DW performance with the incorporation of the well-knitted two-tier user authentication techniques. This approach has ensured data confidentiality with performance enhancement by using middleware and also by supporting the multiple level/tier architecture. This solution includes two parts, security implementation by user authentication at tier one; and security implementation by sending an Automatic Code Engendering (ACE) to user's mobile phone at tier two as shown in Fig 7. The ACE algorithm has a code of six that would be alphanumeric. Three of the code length would be alphabets and the other three would be digits. The first, third and fifth positions would have placed with alphabets, and digits would be placed in the second, fourth and sixth positions.

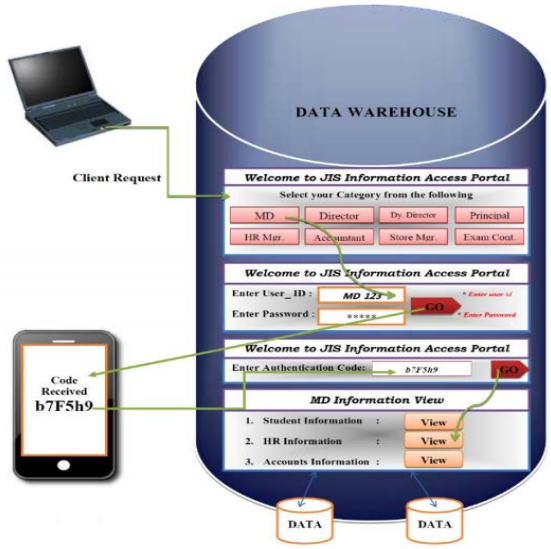

Fig. 7 Process Diagram Illustrating of login for the Entire Process

4. Comparative Analysis

The standard attributes for using the security issues is CIA as published in National Institute of Standards and Technology (NIST) and Federal Information Processing Standard (FIPS). When an approach uses the encryption algorithm to ensure data confidentiality, this cannot resist the attacking in DW systems. For example, as appeared in the DW security approaches, some of them have used the AES algorithm but does not consider the generated key or the shared key for this algorithm. If the attacker is able to get this key, then this approach cannot ensure confidentiality as one of three security issues. This paper presents the evaluation of DW security approaches covered in the two categories:

4.1 Comparison of DW Security Approaches for Confidentiality and Integrity

This subsection draws a comparison of DW security approaches that ensure confidentiality and integrity by considering a set of factors. These factors are the Algorithm Name, the Data Encryption Algorithm Used, the Integrity Type, the Data Environment, the Authorization Type and Tools. Table 1 summarizes a comparison covering DW security approaches of the first category.

As shown in table 1, all the approaches have ensured data confidentiality in DWs using data encryption algorithms. The approaches of [7] [9] and [11] have used the same algorithm named AES but with different types of encryption key sizes. The approaches of [7], [8] and [9] have ensured data integrity named digital certificate or signature. The approaches of [7] and [11] have used the same authorization type named user certification authority.

These approaches have applied their data in different environments such as Microsoft SQL Server 2008-Database contents, SSB shared DW, Hospital Database, Traditional Business Data Warehouse, etc. Also, these approaches used several tools in their experimental studies. For example, the approach of [7] used DPAPI tool to encrypt the service master key in order to help in creating user's certificates. The approach of [8] used set of tools such as Dev-C++ 5.5.3, MySQL 5.0.51a, and OLAP tool. The approach of [9] used TTP Metadata. The approach of [10] used the C-Store tool (open-source column-oriented DBMS) and the crypto library in OpenSSL 0.9.8b. The approach of [11] used the security catalog tool to generate the public and private keys through the RSA algorithm. The approach of [12] used different tools like IBM DB2, Oracle, SQL Server, and Sybase. Finally, the approach of [13] used ASCII and EBCDIC with encryption/decryption operations.

Table 1: Shows the comparison of DW security approaches ensured confidentiality and integrity

Approaches Proposed by	Algorithm Name	Data Encryption Algorithm Used	Integrity type	Data Environment	Authorization Type	Tools
A. Deshmuk and R. Qureshi (2011)[7]	Transparent Data Encryption (TDE) using master database key	AES-128 algorithm	Digital Certificate	Microsoft SQL Server 2008-Database contents	Using master key and creating user certificate	DPAPI (encrypt the Service Master key)
Attasena et al. (2015)[8]	A novel multi-secret sharing approach	Homomo- rphic and incremental algorithms	Inner signature and outer signature	(Star Schema Benchmark) SSB shared DW	Using pseudorandom coefficient by linear equation	Dev-C++ 5.5.3, MySQL 5.0.51a, OLAP tool
H. Kadhem, T. Amagasa, and H. Kitagawa (2009)[9]	Mixed Cryptography Database (MCDB)	Symmetric encryption algorithm	Result analyzer (RA) and query management agent(QMA)	Hospital Database	Using metadata rules	Trusted Third Party (TTP) Metadata
T. Ge and S. Zdonik (2007) [10]	Fast Comparison Encryption (FCE)	FCE as a symmetric encryption algorithm	Not used	Traditional Business Data Warehouse	Not used	C-Store, and Crypto library in OpenSSL 0.9.8b.
G. Chen, K. Chen, and J. Dong (2006)[11]	Novel Database Encryption scheme for enhanced data sharing	CAST_128 and AES 256 algorithms	Not used	Oscar Relational Database Management System (RDBMS).	Certification Authority (CA), using public key and private key	Security Catalog (RSA)
L. Bouganim and Y. GUO (2011) [12]	Hash Security Module Encryption Strategy	State of the art algorithm and a mode of operation	Not used	Relational Database	Using access control policies	IBM DB2, Oracle, SQL Server, and Sybase
M. S. Reddy et. al. (2011)[13]	A Schematic Technique Using Data type Preserving Encryption	DES-64 algorithm	Not used	Multifaceted data warehouse	Not used	DES algorithm with operations in ASCII and EBCDIC

Table 2: Shows the comparison of DW security approaches presented in Table 1 through security considerations

Approaches Proposed by	Security Consideration					
	Access Control	Inference Policy	User Authentication	Accoun-tability	Auditing	Encryption
A. Deshmuk and R. Qureshi (2011)[7]	✔	✔	✔	✔	✔	✔
Attasena et al. (2015)[8]	✔	✔	✔			✔
H. Kadhem, T. Amagasa, and H. Kitagawa (2009)[9]	✔		✔	✔	✔	✔
T. Ge and S. Zdonik (2007) [10]						✔
G. Chen, K. Chen, and J. Dong (2006)[11]	✔					✔
L. Bouganim and Y. GUO (2011) [12]	✔					✔
M. S. Reddy et. al. (2011)[13]		✔				✔

There are general security considerations, which are defined for the security policy of organization. These security considerations contain six attributes such as Access Control, Inference Policy, User Authentication, Accountability, Auditing, and Encryption [5], [19]. As shown in Table 2, all of the approaches applied data encryption to ensure confidentiality in DWs. The approaches of [7], [8], [9], [11], and [12] ensured access control in their systems. The approaches of [7], [8] and [9] ensured user authentication. While the approaches of [7] and [9] ensured accountability and auditing considerations. Finally, we see that the approach of [7] achieved all the security considerations. This approach is the best one because it uses digital certificate or signature. This means

that in future work, we will propose an approach that will apply digital signature in DW environments like the approach of Deshmuk.

4.2 Comparison of DW Security Approaches for Confidentiality and Availability as High-Performance

In this paper, we evaluate and compare the existing DW security approaches that achieve high performance by using middleware. This comparison considers a set of factors such as the Algorithm Name, the Data Encryption Algorithm Used, the Integrity Type, the Data Environment, the Authorization Type, the Middleware Name, and the Middleware Type. Table 3 presents a comparison that covers the proposed DW security approaches of the second category.

Table 3: Shows the comparison of DW security approaches ensured confidentiality and availability to reach high performance

Approaches proposed by	Algorithm Name	Data Encryption Algorithm Used	Integrity type	Data Environment	Authorization Type	Middleware Name	Middleware Type	Tools
R. J. Santos , D. Rasteiro, J. Bernardin, and M. Vieira (2013) [14]	Specific Encryption Solution tailored for DWs (SES-DW	SES-DW, AES-256 algorithms	Not used	Sales DW	Not used	SES-DW Middleware	Cluster Computing	Oracle 11g and Microsoft SQL Server 2008
Khairnar and Patil (2015)[15]	Distributed Data Mining Algorithm (DM)	DM, AES, and (K & C) Kantarcioglu and Clifton algorithms	Not used	KDD Community Datasets	Not used	Site Centralized Server	Cloud computing	KDD community and Java NetBeans
R. J. Santos, J. Bernardino, and M. Vieira (2011)[16]	A Modulus-Based Technique For Data Masking in DW (MOBAT)	MOBAT algorithm	Not used	Sales DW	User access definitions and masking keys in black box	MOBAT-SA	Cluster Computing	Oracle 11g DBMS
Preeti and K. Khatka (2015) [17]	TSFS encryption algorithm for enhancing data security and privacy on WebOS	Transposition-Substitution-Folding-Shifting (TSFS) algorithm	Not used	Company Market	Password Encryption	WebOS Virtual Desktop	Cloud Computing	ASP.Net with Visual C# and SQL Server 2008
R. Chowdhury, P.Chatterjee, P. Mitra, and O. Roy (2014) [18]	Security mechanism for DW performance enhancement using two tier user authentication techniques	Automatic Code Engendering (ACE) Algorithm	Not used	Local Organization Data Warehouse- Glance DW	An auto engend-ered code	Data Warehouse Middleware	Cluster Computing	JIS Information Access Portal

Table 3 shows that all of the approaches have ensured data confidentiality by using different encryption algorithms. Also, these approaches have ensured data availability for high performance in DWs using middleware. The approaches of [14], [16] and [18] used the same middleware type named cluster computing. While the approaches of [15] and [17] used cloud computing middleware type. No one of those approaches could ensure the data integrity in DW systems as hash functions. These approaches applied their data in different environments such as Sales DW, KDD community Datasets, Company Market and Local Organization Data Warehouse-Glance DW. The approaches of [16], [17] and [18] have ensured user authorization in their systems.

Each of these approaches used different tools in the experimental environment. For example, the approach of [14] used Oracle 11g and Microsoft SQL Server 2008. The approach of [15] used the KDD community and Java Net Beans. In the [16], the authors used Oracle 11g DBMS. The approach of [17] used ASP.Net with Visual C# and SQL Server 2008. The approach of [18] used the JIS Information Access Portal. Finally, the DW security approaches of the second category have solved the overhead problem in the network without considering the data integrity (i.e., checking if the data received by the client is changed or not). Therefore, the next paper will propose a new approach to ensure data integrity when using the middleware in order to cover all the security issues in DWs.

5. Conclusion and Future Work

When many organizations have shared partnerships in a DW source, confidential data such as the financial data becomes the main target of attackers. Confidentiality is not enough to ensure a complete security solution in DW

systems. Where the change of the encrypted data during transmission process as attacking leads to incorrect decision for business analysis in DWs. This paper has presented a survey of DW security approaches covered in two categories depending on the CIA security issues. In the first category, some of the approaches have ensured data confidentiality and integrity in DWs. The other category has provided many approaches ensuring data confidentiality and availability without ensuring data integrity. This paper has presented a comparison of DW security approaches for the two categories by using a set of factors. In this paper, we have showed the need for using the digital signature as a new approach, which will be a framework for implementing security issues in DWs. It will ensure the three security issues (Confidentiality, Integrity, Availability CIA) in DW systems in a next research paper.

References

[1] D. Mankad and P. Dholakia, "The Study on Data Warehouse Design and Usage," International Journal of Scientific and Research Publications, ISSN 2250-3153 Volume 3, Issue 3, pp. 1-5, March 2013.

[2] A. Sing and N. Umesh, "Implementing Log Based Security in Data Warehouse," International Journal of Advanced Computer Research, ISSN (print): 2249-7277, ISSN (online): 2277-7970), Volume-3, Number-1, Issue-8, March 2013.

[3] R.J. Santos, J. Bernardino and M. Vieira, "Balancing Security and Performance for Enhancing Data Privacy in Data Warehouses," Proceedings of the 10th International Conference on Trust, Security and Privacy in Computing and Communications, Changsha, DOI 10.1109/TrustCom.2011.33, pp. 242-249, IEEE, 16-18 November 2011.

[4] S. Konda and R. More, "Augmenting Data Warehouse Security Techniques - A Selective Survey," International Research Journal of Engineering and Technology (IRJET), e-ISSN: 2395 -0056, p-ISSN: 2395-0072, Volume 02, Issue 03, pp. 2209- 2213, June 2015.

[5] I. Basharat, F, Azam and A.W. Muzaffar, "Database Security and Encryption: A Survey Study," International Journal of Computer Applications (0975-888), Volume 47, No.12, pp. 28-34, June 2012.

[6] S. Aleem, L. F. Capretz, F. Ahmed, "Security Issues in Data Warehouse," Recent Advances in Information Technology, ISBN: 978-1-61804-264-4, pp. 15-20, Canada 2015.

[7] A. Deshmuk and R. Qureshi, "Transparent Data Encryption-Solution for Security of Database Contents," International Journal of Advanced Computer Science and Applications (IJACSA), Vol. 2, No.3, pp. 25-28, March 2011.

[8] V. Attasena, N. Harbi and J. Darmont, "A novel multi-secret sharing approach for secure data warehousing and on-line analysis processing in the cloud," International Journal of Data Warehousing and Mining (IJDWM), Vol. 11, No. 2, pp. 22-43, 2015.

[9] H. Kadhem, T. Amagasa and H. Kitagawa, "A Novel Framework for Database Security based on Mixed Cryptography," Proceedings of the 4th International Conference on Internet and Web Applications and Services, Venice/Mestre, pp. 163-170, IEEE, 24-28 May 2009.

[10] T. Ge and S. Zdonik, "Fast, Secure Encryption for Indexing in a Column-Oriented DBMS," Proceedings of the 23rd International Conference on Data Engineering (ICDE), Istanbul, pp. 676-685, IEEE, May 2007.

[11] G. Chen, K. Chen and J. Dong, "A Database Encryption Scheme for Enhanced Security and Easy Sharing," Proceedings of the 10th International Conference on Computer Supported Cooperative Work in Design(CSCWD), Nanjing, pp. 1-6, IEEE, 3-5 May 2006.

[12] L. Bouganim and Y. GUO, "Database Encryption" "Encyclopedia of Cryptography and Security," H. van Tilborg and S. Jajodia (Ed.), Springer Science+Business Media, LLC, 2nd Edition, ISBN: 978-1-4419-5905-8, pp. 307-312, 2011.

[13] M. S. Reddy, M. R. Reddy, R. Viswanath, G.V. Chalam, R. Laxmi and Md. A. Rizwan, "A Schematic Technique Using Data type Preserving Encryption to Boost Data Warehouse Security," International Journal of Computer Science Issues(IJCSI), Vol. 8, Issue 1, ISSN (Online): 1694-0814, pp. 460-465, January 2011.

[14] R. J. Santos, D. Rasteiro, J. Bernardino and M. Vieira, "A Specific Encryption Solution for Data Warehouses," Proceedings of International Conference on Database Systems for Advanced Applications. Springer Berlin Heidelberg, pp. 84-98, April 2013.

[15] P. B. Khairnar and D.V. Patil, "Implementing Security In Distributed Data Mining Approaches," International Journal of Engineering, Business and Enterprise Applications (IJEBEA), International Association of Scientific Innovation and Research (IASIR), ISSN (Print): 2279-0020, pp. 148-152, 2015.

[16] R. J. Santos, J. Bernardino, and M. Vieira, "A data masking Technique for Data Warehouses," Proceedings of 15th International Conference on Database Engineering and Applications Symposium (IDEAS), ACM, pp. 61-69, 21 - 27 September, 2011.

[17] Preeti and K. Khatka, "Enchancing Data Security And Privacy On Webos Using TSFs," International Journal of Computer Engineering And Electronics Technology (IJCEET), Vol. 1, Issue 1, pp. 1-5, 2015.

[18] R. Chowdhury, P. Chatterjee, P. Mitra and O. Roy, "Design and Implementation of Security Mechanism for Data Warehouse Performance Enhancement Using Two Tier User Authentication Techniques," International Journal of Innovative Research in Science, Engineering and Technology, Volume3, Special Issue 6, February 2014.

[19] T. Priebe and G. Pernul, "Toward OLAP Security Design –Survey and Research Issues," Proceedings of the 3rd ACM International Workshop on Data warehousing and OLAP, McLean, Virginia, USA, pp. 33-40, 6-11 November 2000.

Mayada J. ALMeghari B.Sc. of Computer Information Systems, Faculty of Technology and Applied Sciences (FTAS), Al-Quads Open University, Palestine 2002. M.Sc. of Computer Information Systems, Faculty of Information Systems & Technology, Arab Academic for Banking & Financial Sciences University, Jordan 2006. She has been an academic lecturer at Palestine Technical College-Deir El-Balah since 2007. This paper was presented as one of published work to obtain PhD. degree in Information Systems from Faculty of Computers and Information, Cairo University, Egypt.